

UNDER DEVELOPMENT

Raging bull

As demands are placed on him left, right and centre, the normally mild-mannered David Robinson starts to feel the pressure – and blows his top

I lost my cool this month – assuming I had any to lose, of course. I'm normally quite good under pressure, but there's pressure and there's *pressure*. The problem, as always, is the custards. Three or four projects have been dragging on for ages while we've waited for decisions or action from custards. What usually happens is that the project is stalled for weeks, then they come up with something and expect you to take it to the next stage in a couple of days. And they moan like hell if you don't.

And it's not just the custards. These days there's the customer's customer. And the customer's customer's customer. And in one extreme case the customer's customer's customer enters the equation, as our customer moves things for his customer who sells to his customer abroad who turns out to be an agent who never actually handles the items as they go directly from our customer to the agent's customer. And they all want to know stuff about the items. Confused? Who wouldn't be?

TOUGH LUCK

Murphy's law being what it is, three projects became 'unbunded' at the same time, just before I'm due a week's holiday and then flying out to the USA, where another overdue project awaits. To compound the problem, Keef was off sick. If I find the smart arse who said "when the going gets tough..." I'll kick him. Damn, there goes my cool again!

The worst one to deal with is the project with four levels of customer, as there are also four levels of contact, too. There's my customer, the programmers at my customer's customer's office, their managers and, finally, the big shots at the holding company that owns the customer's customer. These are the faceless ones who issue all the diktats regarding the timing of the project but do so without any consideration for what, where, when, how and by whom. It's just "finish it by the end of March – or else" and it's not negotiable, not least because I can't gain access to these people to negotiate with



them. There's just the potential for a whole queue of brassed off people between the two of us.

I got out of working in big companies because of the politics and having somebody higher up the corporate pole imposing impossible demands with an 'or else' either implied or explicitly enunciated. When your daily bread comes from a variety of customers, in theory you can tell any one of them to bog off without facing starvation, because other customers still want your services. In practice, however, it goes against the grain to blow out hard-won clients unless they're extremely obnoxious. We've only done it twice in 25 years and one of those came back after two years and (almost) begged for forgiveness.

GO WITH THE OVERFLOW

I issued lots of warnings about the impossible time scale and settled down to long hours of coding and manual writing. I got to a point where the main program, which is to run on a PPC-powered PDA/scanner, was nearly finished. I compiled the latest set of changes, downloaded the program to the PDA and started another test run. And the program blew up, reporting a stack overflow error. The changes don't do anything that would involve the CPU stack. What could be wrong?

I'd read somewhere in the compiler documentation that there is a limit on the number of objects the language can handle, but although it's a biggish program for a

PDA, with around 2,000 lines, I didn't think I'd bust the limit.

After lots of worrying about how to count the number of objects I was using, I decided to run the code through the real-time debugger, which is what I should have done straight away. Early in the program timeline I got to some code that (approximately) went:

```
function showmainmenu
do_this
do_that
do_somethingelse // just
preparatory code
show mainmenu
end;
```

I'd been tidying up some of the code I'd written earlier and the last line should have read:

```
show frmmainmenu
```

so at this point the program was just calling the same function over and over again, which led to a stack overflow. A stupid beginner's mistake, compounded by going off on the wrong fault-finding track, all because I was trying to do too much in a short space of time. Thankfully, it was easily fixed.

On delivering the program we did some tests on the warehouse floor. The PDA downloads a shipment manifest from a remote mainframe, sorts the items into order and then, when you scan the barcodes on the items, it checks them into stock. It's similar to the program described in last month's

Under Development, but with more features. So we downloaded the manifest and started scanning. However, the program reported an "item not in shipment" error for half the items in the shipment, which meant that the sender had sent an item not in the manifest. That does happen sometimes, but never for half a shipment. What was wrong?

We checked the data printed under the barcodes against what the scanner had read, and they matched – exactly. Then we checked the serial numbers on the barcodes against a printout of the digital manifest. Some read, for example, manifest E765B982005 barcode E765B982005, which is fine, but some read manifest E765B982006 barcode E765B982006, which to a human being is the same thing but to a computer is definitely not. It turns out that the customer's customer's software doesn't have any rules for verifying serial number input but always prints barcodes in upper-case characters only.

STROPPY SECONDS

Meanwhile, Mr Stroppey, who works for another customer, logged on to his company system and installed some software that's causing problems. He's done this before and been asked not to as it's contrary to the terms of the support agreement but, for some reason, his boss insisted that Mr Stroppey be given the admin password.

I blew my top and, determined to stop this happening again, logged on to their system and changed the admin password. The next morning all hell broke loose as various people couldn't get into this and that, and it was all life and death stuff. Several bits of application software had been installed in such a way that they could be run only by administrator. This is a bad idea. Patrick's not happy with me for causing all this work but, observing my lack of cool, he keeps quiet. ☹

David Robinson
Software and systems
developer

davidr@computershopper.co.uk

